

 <p>UCG Univerzitet Crne Gore</p>	<p>UNIVERZITET CRNE GORE ELEKTROTEHNIČKI FAKULTET</p>	
STUDIJSKI PROGRAM:	<i>ELEKTRONIKA, TELEKOMUNIKACIJE I RAČUNARI</i>	
PREDMET:	<i>PROJEKTOVANJE DIGITALNIH SISTEMA</i>	
FOND ČASOVA:	<i>3 + 0 + 1</i>	

LABORATORIJSKA VJEŽBA

NAZIV:	<i>REALIZACIJA UPROŠĆENOG MODELA SAOBRAĆAJNOG SEMAFORA</i>
<p>CILJEVI VJEŽBE:</p> <ul style="list-style-type: none"> - praktičan rad sa SPARTAN 3E Starter kit razvojnom platformom i Xilinx ISE razvojnim okruženjem, - povezivanje teorijske osnove (predavanja) sa praktičnom realizacijom, - upoznavanje sa <i>behavioral</i> modelovanjem, - detaljnije upoznavanje sa postupcima projektovanja složenijih digitalnih sistema i načinom opisivanja automata u Verilogu, - programiranje PROM-a na Spartan 3E Starter kit platformi. 	
<p>POTREBAN PRIBOR:</p> <ul style="list-style-type: none"> - pribor za pisanje. 	

IME I PREZIME: _____.

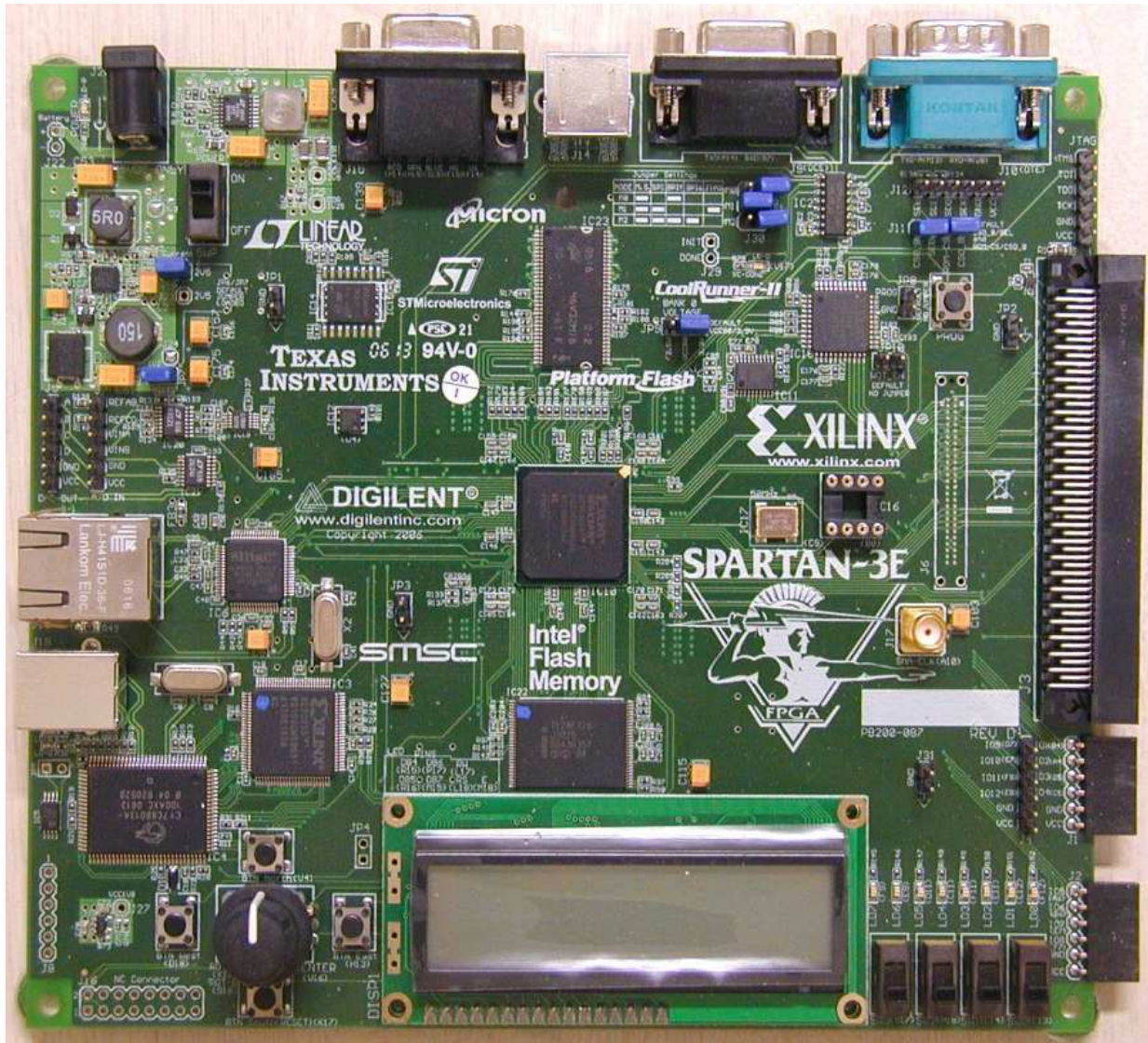
BROJ INDEKSA: _____.

BROJ POENA:	
OVJERAVA:	
DATUM:	

1. APARATURA

Na raspolaganju su sljedeći uređaji i oprema:

- SPARTAN 3E Starter Kit razvojna platforma,
- računar,
- napajanje,
- USB kabal,
- Xilinx ISE Design Suite v.14.7 razvojno okruženje.



Slika 1.1: Izgled SPARTAN 3E Starter Kit razvojne platforme

2. TEORIJSKA OSNOVA LABORATORIJSKE VJEŽBE

Automat (eng. **FSM** – *Finite State Machine* – mašina sa konačnim brojem stanja) predstavlja sekvencijalno kolo, koje ima određeni broj stanja u kojima se može naći i pravila saglasno kojima može preći iz jednog stanja u drugo. Prelazak iz jednog stanja u drugo se najčešće dešava na osnovu vrijednosti nekih spoljašnjih parametara (signala), pa automat gotovo uvijek ima ulazne signale. Takođe, automat ima određeni broj izlaznih signala preko kojih saopštava informaciju karakterističnu za stanje u kome se nalazi.

Automati se, tokom projektovanja, najčešće predstavljaju grafički, pomoću grafova. Postoje dva tipa automata: murov (Moore) i milijev (Mealy) automat. Kod murovog automata izlaz zavisi isključivo od stanja u kome se nalazi, dok kod milijevog automata izlaz zavisi od stanja i od trenutnog ulaza.

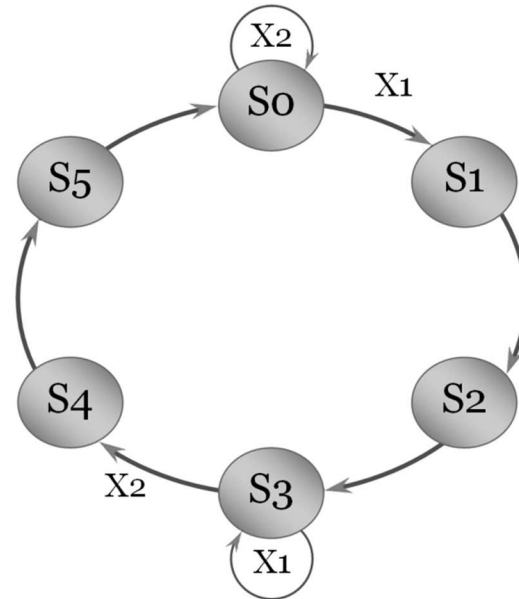
U ovoj vježbi će biti realizovan semafor koji uređuje raskrnicu sa dva saobraćajna pravca. Na svakom od pravaca se nalazi senzor koji detektuje prisustvo vozila na tom pravcu. Sensori su označeni ulaznim promjenljivima X1 i X2. Semafor će propuštati vozila po jednom pravcu sve dok po tom pravcu ima vozila (odgovarajući senzor je aktivan, odnosno ulazna promjenljiva jednaka jedinici). Po tom pravcu će biti upaljeno zeleno svjetlo, a po drugom pravcu crveno svjetlo. Do promjene svjetlosne signalizacije dolazi ako po trenutno aktivnom pravcu više nema vozila (ulazna promjenljiva jednaka nuli), a po drugom pravcu ima vozila na čekanju (ulazna promjenljiva jednaka jedinici). Prilikom promjene svjetlosne signalizacije neophodno je da se nakon zelenog svjetla neko vrijeme pojavi žuto, pa tek onda crveno svjetlo. Slično, nakon crvenog svjetla neko vrijeme treba da bude upaljeno crveno i žuto, a tek onda zeleno svjetlo.

Za realizaciju opisanog semafora pogodno je koristiti murov automat kod koga eventualna promjena ulaznog signala (zbog prolaska vozila) neće uticati na izlazne vrijednosti (svjetla na semaforu). Svako svjetlosnoj kombinaciji treba pridružiti po jedno stanje. Kao što se iz tabele 1 može vidjeti, za realizaciju ovakvog semafora potrebno je 6 stanja (označenih sa S0 do S5).

Tabela 1. Ciklus rada semafora

STANJE	IZLAZI
S0	na prvom semaforu svijetli crveno, a na drugom zeleno
S1	na prvom semaforu svijetli crveno, a na drugom žuto
S2	na prvom semaforu svijetli crveno-žuto, a na drugom crveno
S3	na prvom semaforu svijetli zeleno, a na drugom crveno
S4	na prvom semaforu svijetli žuto, a na drugom crveno
S5	na prvom semaforu svijetli crveno, a na drugom crveno-žuto

Dijagram stanja odgovarajućeg automata prikazan je na slici 2.1. Prelazak iz stanja S_0 u stanje S_1 se vrši samo ako po pravcu 2 više nema vozila ($X_2=0$) a po pravcu 1 ima ($X_1=1$). Započeta promjena svjetlosne signalizacije se mora završiti do kraja pa se zato iz stanja S_1 u stanje S_2 , kao i iz stanja S_2 u stanje S_3 , prelazi bezuslovno (bez obzira na vrijednosti ulaznih promjenljivih). Slično tome, prelazak iz stanja S_3 u stanje S_4 se vrši samo ako po pravcu 1 više nema vozila ($X_1=0$) a po pravcu 2 ima ($X_2=1$). Započeta promjena svjetlosne signalizacije se mora završiti do kraja pa se zato iz stanja S_4 u stanje S_5 , kao i iz stanja S_5 u stanje S_0 , prelazi bezuslovno.



Slika 2.1: Dijagram stanja uprošćenog saobraćajnog semafora

2.1 Verilog kod za realizaciju semafora

```

`define TRUE 1'b1
`define FALSE 1'b0
`define CRVENO 4'd1
`define ZUTO 4'd2
`define ZELENO 4'd4

// definicije stanja: prvi           drugi
`define S0 4'd0 // crveno           zeleno
`define S1 4'd1 // crveno           zuto
`define S2 4'd2 // crveno i zuto    crveno
`define S3 4'd3 // zeleno           crveno
`define S4 4'd4 // zuto             crveno
`define S5 4'd5 // crveno           crveno i zuto
// kasnjenja
  
```

```
`define ZUTO_CRVENO 2
`define CRVENOZUTO_ZELENO 2

module semafor(prvi, drugi, X1, X2, clock, clear);
output [3:0] prvi, drugi; // 3 moguca svijetla (C,z,Z)
reg [3:0] prvi, drugi;
input X1, X2; // ako je TRUE ima vozila na putu
input clock, clear;

reg [31:0] counter;
// pomocne promjenljive (za stanja automata)
reg [3:0] stanje;
reg [3:0] sljedece_stanje;

// semafor pocinje u stanju S0
initial // inicijalizacija
begin
    stanje = `S0;
    sljedece_stanje = `S0;
    drugi = `ZELENO;
    prvi = `CRVENO;
end

always @(posedge clock) // promjena stanja se vrsi na pozitivnoj ivici takta
begin // ovdje se takt CLK_50M dijeli sa 100 miliona => 2s
    if(counter==32'd100_000_000) begin
        counter <= 32'd0;
        stanje = sljedece_stanje;
    end
    else begin
        counter<=counter+1;
    end
end

// Izracunati vrijednost svjetlosnih signala
always @(stanje)
begin
    case(stanje)
        `S0:   begin
                prvi = `CRVENO;
                drugi = `ZELENO;
            end
        `S1:   begin
                prvi = `CRVENO;
                drugi = `ZUTO;
            end
        `S2:   begin
                prvi = `CRVENO + `ZUTO;
            end
    endcase
end
```

```

                                drugi = `CRVENO;
                                end
`S3: begin
                                prvi = `ZELENO;
                                drugi = `CRVENO;
                                end
`S4: begin
                                prvi = `ZUTO;
                                drugi = `CRVENO;
                                end
`S5: begin
                                prvi= `CRVENO;
                                drugi= `CRVENO + `ZUTO;
                                end
                                end
                                endcase
                                end
// automat
always @(stanje or clear or X1 or X2)
begin
    if (clear)
        sljedece_stanje = `S0;
    else
        case (stanje)
            `S0: begin
                    if(X1 && !X2) sljedece_stanje = `S1;
                    else sljedece_stanje = `S0;
                end
            `S1: begin
                    sljedece_stanje = `S2;
                end
            `S2: begin
                    sljedece_stanje = `S3;
                end
            `S3: begin
                    if(X2 && !X1) sljedece_stanje = `S4;
                    else sljedece_stanje = `S3;
                end
            `S4: begin
                    sljedece_stanje = `S5;
                end
            `S5: begin
                    sljedece_stanje = `S0;
                end
            default: sljedece_stanje = `S0;
        endcase
    end
endmodule

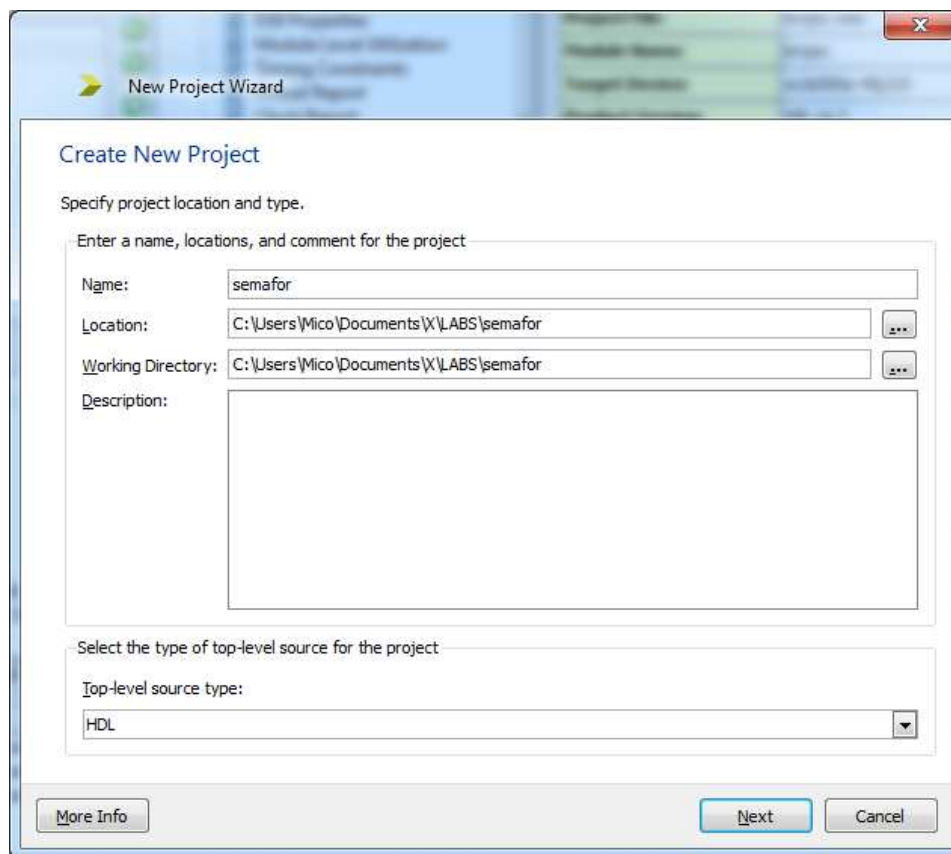
```


3. ZADACI LABORATORIJSKE VJEŽBE

Kreiranje dizajna

Pokrenuti *ISE Project Navigator* i napraviti novi projekat.

1. **Start**→**Programs**→**Xilinx Design Tools**→ **ISE Design Suite 14.7**→**ISE Design Tools** → **Project Navigator**
2. U dobijenom prozoru treba izabrati **File**→**New Project**. Otvoriće se **New Project Wizard** (slika 3.1).

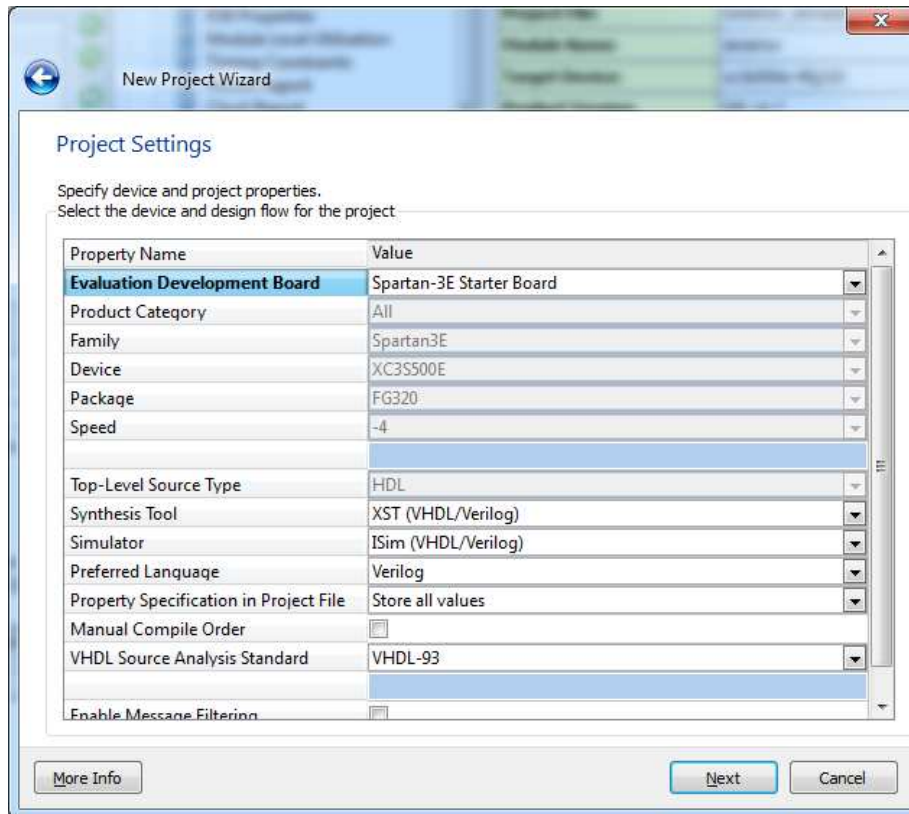


Slika 3.1: Prozor *New Project Wizard*

Dati ime novom projektu (npr. **semafor**), a zatim odrediti mjesto na hard disku gdje želite da vaš projekat i njegovi prateći fajlovi budu sačuvani (...)=**Browse**.

3. <Next>

Pojaviće se prozor prikazan na slici 3.2.

Slika 3.2: Prozor *Device Properties*

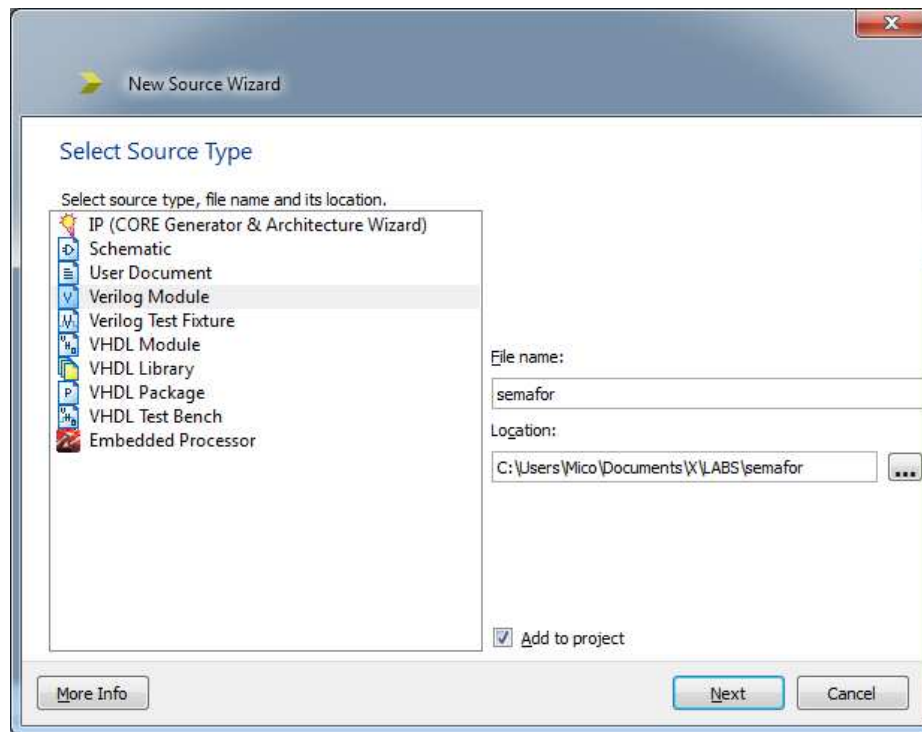
Pojaviće se dijalog za odabir uređaja i toka dizajna i treba odabrati sledeća podešavanja:

Device Family: **Spartan3E**
 Device: **xc3s500E**
 Package: **fg320**
 Speed Grade: **-4**
 Synthesis Tool: **XST (VHDL/Verilog)**
 Simulator: **ISim (VHDL/Verilog)**
 Preferred Language: **Verilog**

Napomena: Može se pod **Evaluation Development Board** izabrati **Spartan-3E Starter Board** i tada bi se automatski podesile opcije za FPGA čip.

4. <Next>, pa <Finish>
5. Izabrati **Project**→**New Source**.

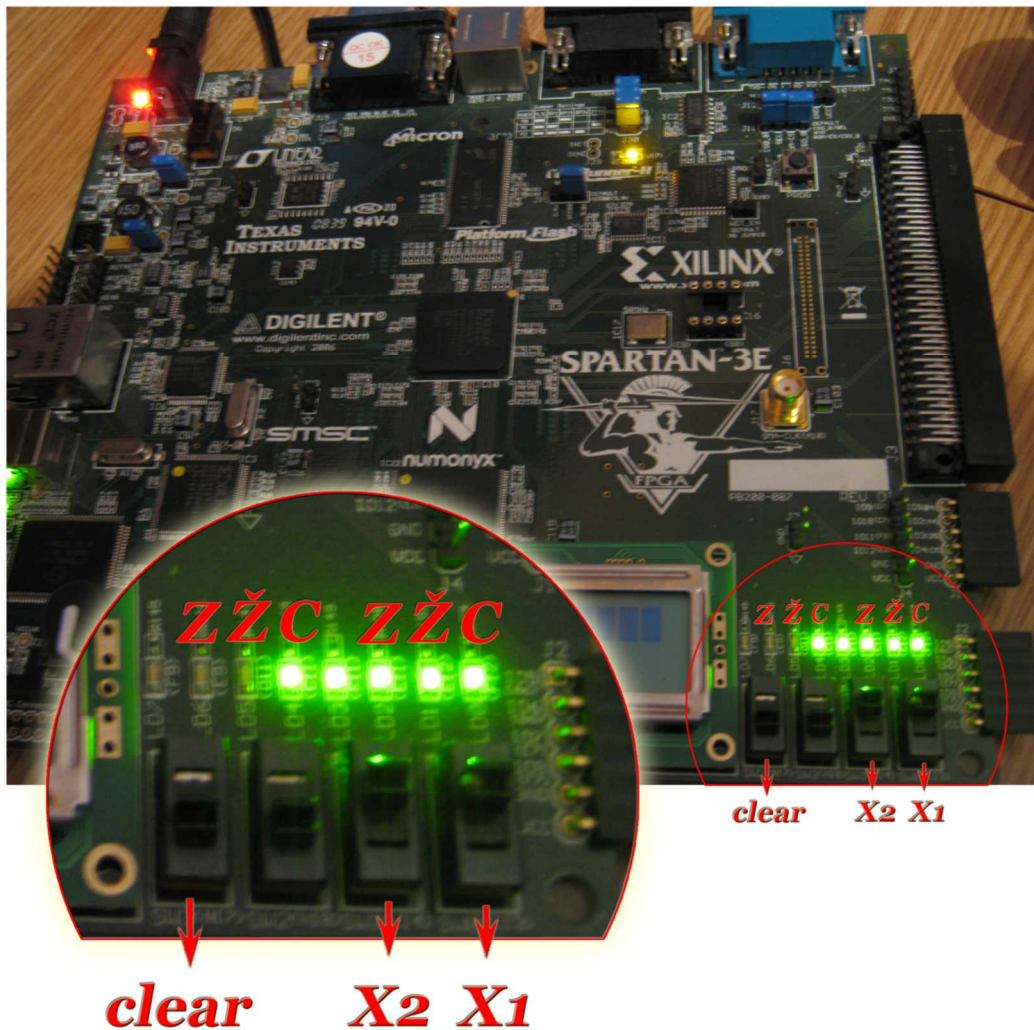
Pojaviće se sledeći prozor (**New Source Wizard** - slika 3.3) u kome treba izabrati opciju **Verilog Module**, dati ime fajlu (npr. **semafor**) i pritisnuti <Next>.

Slika 3.3: Prozor *New Source Wizard*

6. Sledeći prozor (*Define module*) namijenjen je unosu naziva portova i njihovih smjerova. Popunjavanje ovog prozora se može preskočiti jer se odgovarajući verilog kod može kasnije ručno unijeti. Zato izabrati <Next> i <Finish>. Na pitanje da li želite da kreirate folder odgovoriti sa <Yes>.
7. Dobijeni okvir za kreiranje modula dopuniti funkcionalnošću modula saglasno kodu prikazanom u poglavlju 2.1.
8. Snimiti izmjene u fajlu: **File**→**Save**

Implementacija

1. Prvi korak je pridruživanje odgovarajućih pinova FPGA čipa signalima unutar našeg dizajna. Ulazne signale *X1*, *X2* i *clear* ćemo povezati na prekidače koji se nalaze na Spartan 3E Starter Kit razvojnoj platformi, a izlazne signale *prvi* i *drugi* na LED (slika 3.4). U tu svrhu možemo koristiti program **PlanAhead**, kao što je to prikazano u laboratorijskoj vježbi „Architecture Wizard i PACE Lab (PlanAhead)“. Međutim, ovdje ćemo isti zadatak obaviti na drugačiji način.



Slika 3.4: Položaj prekidača i LED koji se koriste u ovoj vježbi

2. U **Design View** prozoru selektovati opciju **Implementation**, a u **Design Hierarchy** prozoru selektovati semafor.
3. Izabrati **Project** → **New Source**, u novom prozoru selektovati **Implementation Constraints File** i dati mu ime (npr. **constraints**) (slika 3.5). Potom izabrati <Next>, <Finish>
4. Upravo kreirani fajl, uz pomoć dokumenta *Spartan-3E Starter Kit Board User Guide*, unijeti kod kojim definišemo povezivanje signala unutar dizajna sa pinovima FPGA čipa:

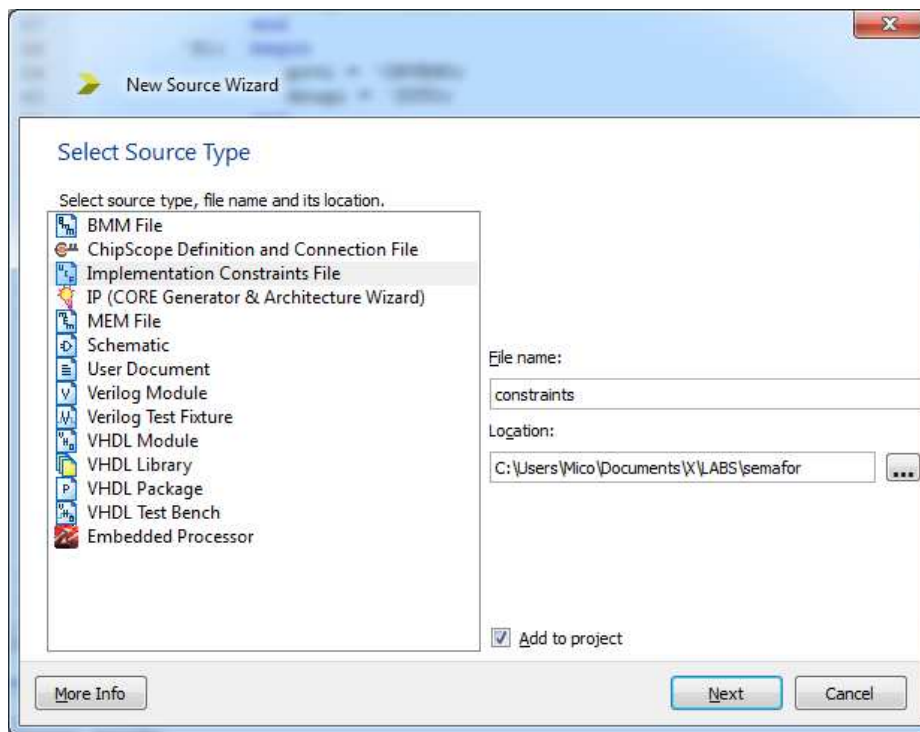
```
NET "clock" LOC = "C9" | IOSTANDARD = LVCMOS33;
NET "clock" PERIOD = 20.0ns HIGH 40%;
```

```
NET "prvi<0>" LOC = "F12" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
NET "prvi<1>" LOC = "E12" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
NET "prvi<2>" LOC = "E11" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
NET "prvi<3>" LOC = "F11" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
```

```
NET "drugi<0>" LOC = "C11" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
NET "drugi<1>" LOC = "D11" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
NET "drugi<2>" LOC = "E9" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
NET "drugi<3>" LOC = "F9" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
```

```
NET "X1" LOC = "L13" | IOSTANDARD = LVTTTL | PULLUP ;
NET "X2" LOC = "L14" | IOSTANDARD = LVTTTL | PULLUP ;
NET "clear" LOC = "N17" | IOSTANDARD = LVTTTL | PULLUP ;
```

5. Snimiti unijete izmjene u *constraint* fajl: **File**→**Save**
6. U **Processes** prozoru dva puta kliknuti na **Generate Programming File** (pri čemu su i dalje selektovane stavke iz koraka 2) i sačekati da se proces završi.
7. Uključiti Spartan-3E Starter Board razvojnu platformu i povezati je sa računarom pomoću USB kabela.

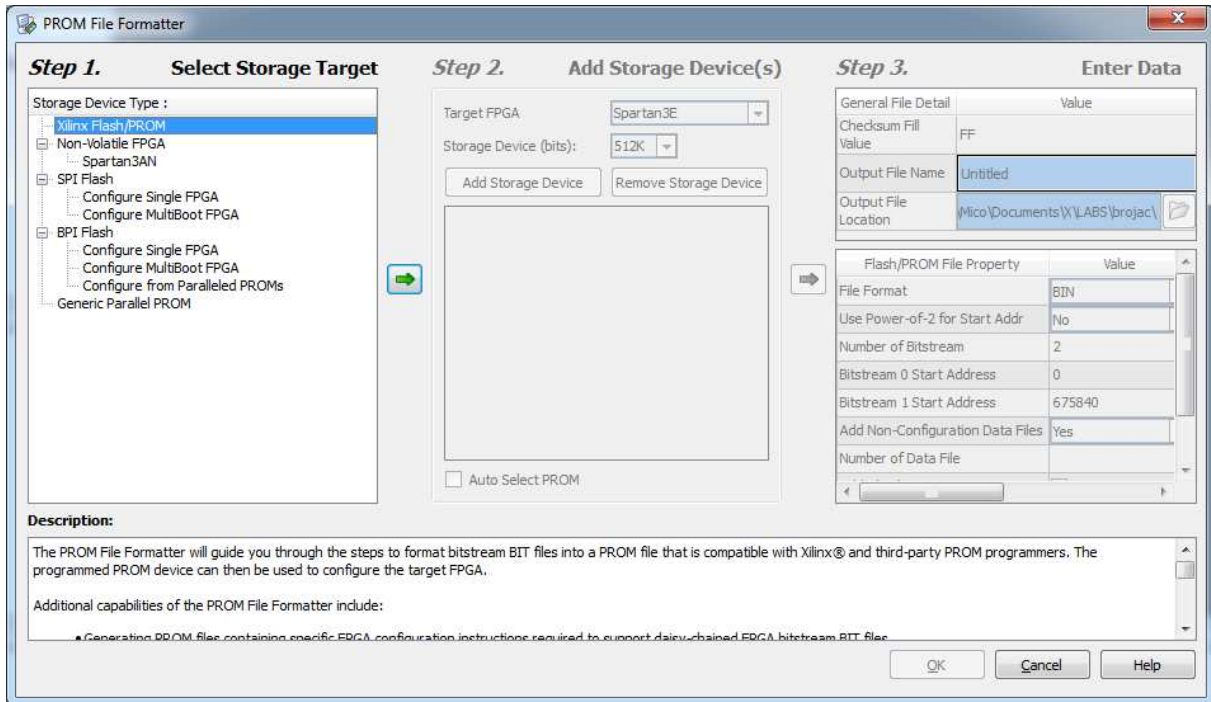


Slika 3.5: Kreiranje *constraints* fajla

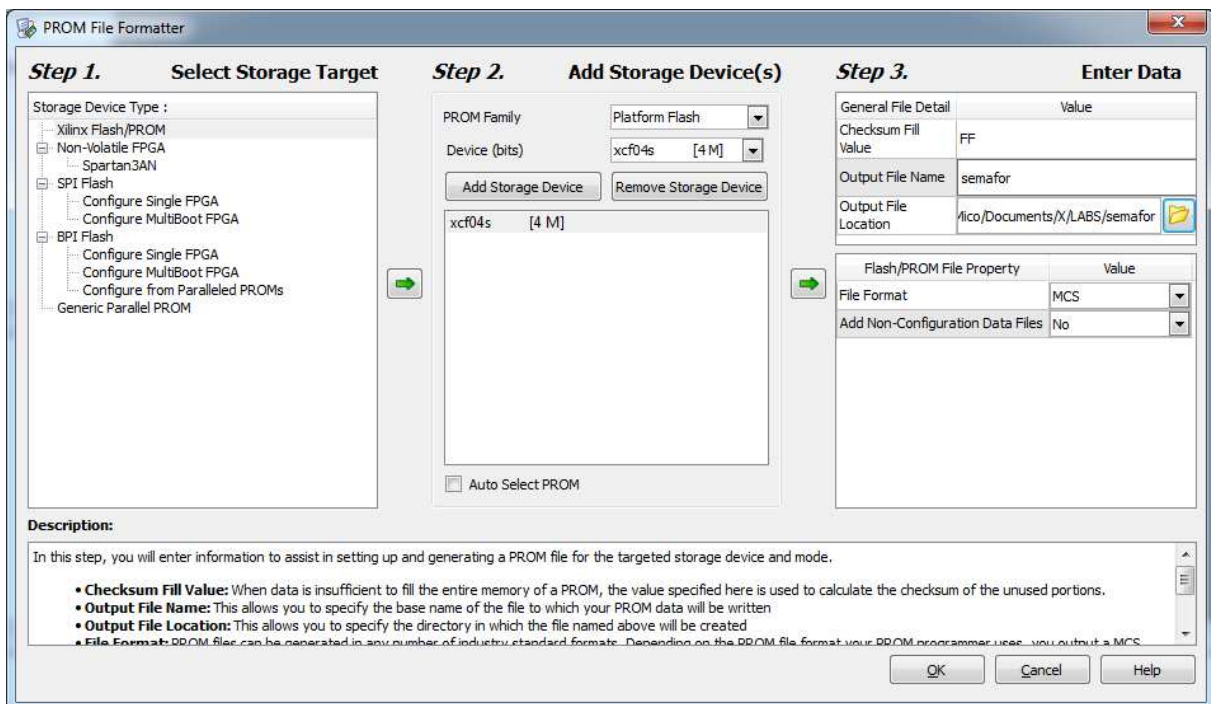
Generisanje Flash PROM fajla

1. U **Processes** prozoru raširiti **Configure Target Device** i dva puta kliknuti na **Generate Target PROM/ACE File**.
2. Kada se pokrene program **iMPACT** u prozoru **iMPACT Flows** dva puta kliknuti na **Create PROM File**. Pojaviće se **PROM File Formatter** prikazan na slici 3.6. U prozoru **Storage Device Type** (Step 1) izabrati **Xilinx Flash/PROM** i pritisnuti zelenu strelicu desno od prozora. Aktiviraće se prozor **Step 2** u kome treba izabrati **PROM Family: Platform Flash** i **Device(bits): xcf04s [4M]** (ovo je Flash memorija koja se nalazi na platformi). Nakon toga

pritisnuti dugme **Add Storage Device**. Označiti dodati uređaj (xcf04s) i pritisnuti zelenu strelicu koja se nalazi desno od prozora **Step 2**. Time je aktiviran prozor **Step 3** u kome treba dati ime izlaznom fajlu (npr. **semafor**). Važno je provjeriti da pod **File Format** stoji **MCS**. Konačan izgled **PROM File Formatter**-a prikazan na slici 3.7. Izabrati **<OK>** i **<OK>**.

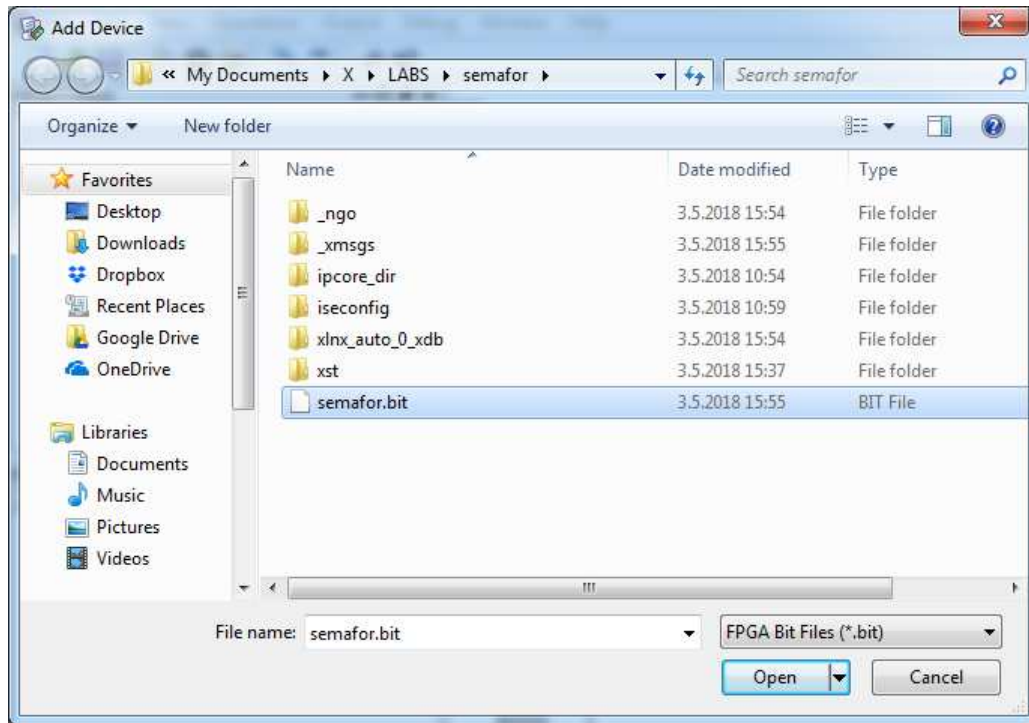


Slika 3.6: PROM File Formatter



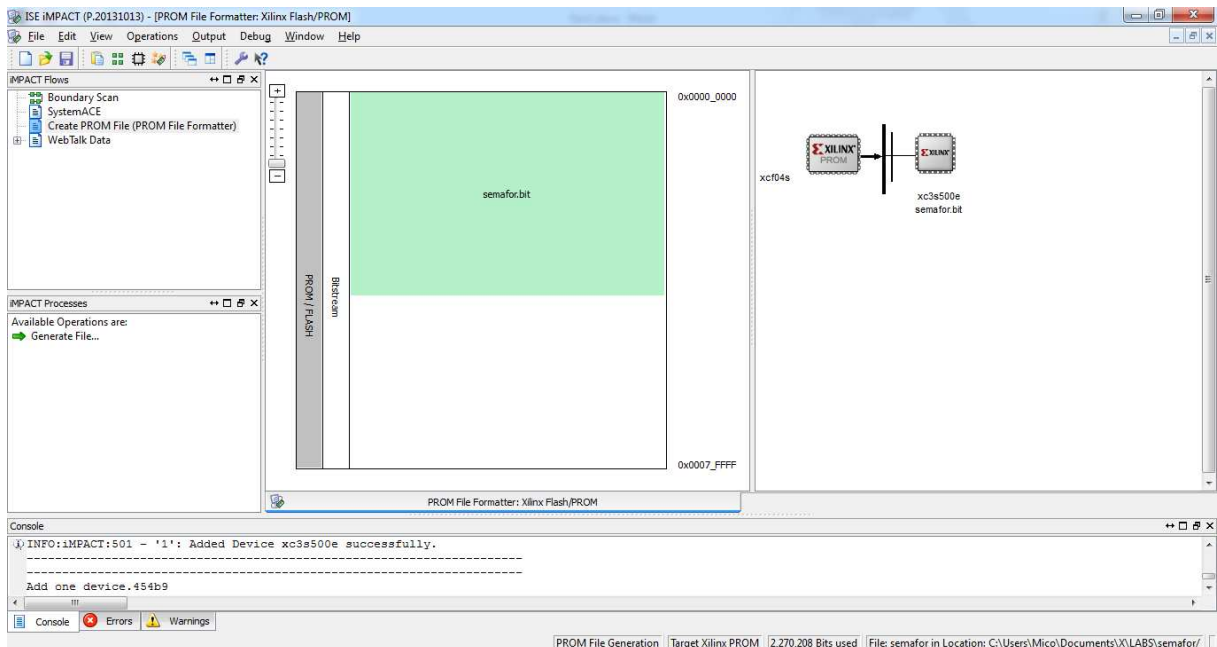
Slika 3.7: PROM File Formatter sa podešenim opcijama

3. Pojaviće se prozor za izbor **.bit** fajla u kome se nalazi željeni dizajn (slika 3.8). Izabrati fajl **semafor.bit** i **<Open>**.



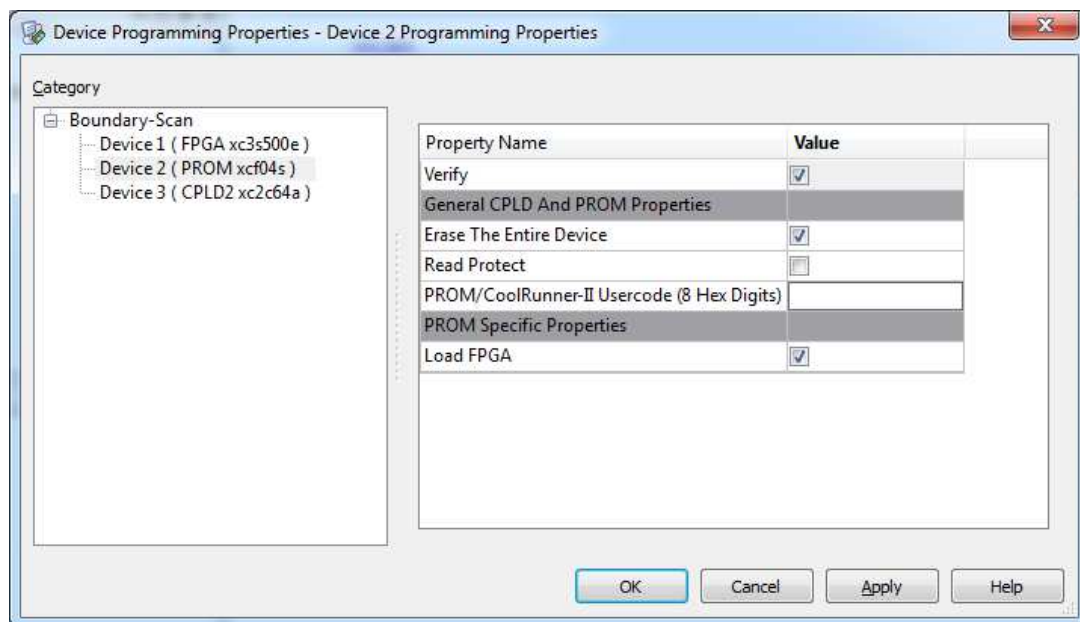
Slika 3.8: Izbor fajla sa dizajnom

4. Na pitanje „**Would you like to add another device file to Revision: 0**“ odgovoriti sa **<No>** i potom pritisnuti **<OK>**. Izgled ekrana prikazan je na slici 3.9.



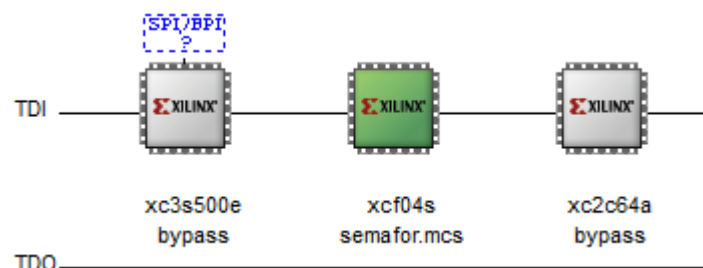
Slika 3.9: PROM File Formatter nakon izbora fajla sa dizajnom

5. U prozoru **iMPACT Processes** dva puta kliknuti na **Generate File**. Proces se završava sa porukom **Generate Succeeded**.
6. Povezati Spartan-3E platformu USB kablom sa računarom, a zatim platformu uključiti.
7. Dva puta kliknuti na **Boundary Scan** u **iMPACT Flows** prozoru. Nakon toga se desnim tasterom miša klikne na praznu bijelu površinu na kojoj piše „**Right click to Add Device or Initialize JTAG chain**“ i izabere se opcija **Initialize Chain**. Na pitanje „**Do you want to continue and assign configuration file(s)**“ odgovoriti sa **<YES>**.
8. Kada se pojavi dijalog **Assign New Configuration File**, za **xc3s500e** (prvi uređaj u JTAG ‘lancu’) se klikne **<Bypass>**.
9. Za drugi uređaj (**xcf04s**) se izabere fajl **semafor.mcs**, a za treći uređaj se klikne **<Bypass>**.
10. Kada se pojavi prozor **Device Programming Properties** selektovati **Device 2 (PROM xcf04s)**, aktivirati opcije **Verify**, **Erase The Entire Device** i **Load FPGA** (slika 3.10) i pritisnuti **<OK>**.



Slika 3.10: PROM File Formatter nakon izbora fajla sa dizajnom

11. Izgled radnog prozora prikazan je na slici 3.11.



Slika 3.11. JTAG lanac sa dodijeljenim konfiguracionim fajlom

12. Kliknuti desnim klikom na ikonicu čipa **xcf04s** koji je označen zelenom bojom (ako nije označen prvo ga selektovati lijevim tasterom miša) u radnom prozoru i odabrati **Program**.
13. Promjenom položaja prekidača i posmatranjem LE dioda provjeriti rad semafora.

Napomena: da bi se FPGA ispravno konfigurisao potrebno je podesiti konfiguracioni mod, odnosno izabrati odakle FPGA preuzima konfiguraciju. Uputstvo za podešavanje konfiguracionog moda se nalazi u fajlu „**Spartan-3E Starter Kit Board User Guide**“, stranice 25 do 27.

4. ZAKLJUČAK